## AMENDMENTS TO THE CLAIMS

Kindly amend claims 67 and 79 as shown in the following listing of claims. The listing of claims will replace all prior versions, and listings, of claims in the application:

### Listing of Claims

1. (previously presented) An integrated circuit functioning as a network interface adapter, comprising:

   a plurality of media access controllers (MACs), for transceiving packets;

   a local bus interface, for performing addressed data transfers on a local bus coupled thereto;

   a bus router, for performing transport layer operations between said plurality of MACs and said local bus interface;

   a memory, shared by said plurality of MACs, said local bus interface, and said bus router, for buffering data received thereby; and

   a transaction switch, coupled to each of said memory, said plurality of MACs, said local bus interface, and said bus router, for switching data and transactions therebetween;

   wherein said bus router is configured to write a packet header into said memory via said transaction switch along with addressed data stored in said memory by said local bus interface to create a packet;

   wherein said local bus interface is configured to read a payload portion of a packet stored in said memory and to transmit said payload portion on said local bus coupled thereto;

   wherein said payload portion is located in said memory at an offset specified in a transaction posted by said bus router to said local bus interface via said transaction switch.

2. (previously presented) The integrated circuit of claim 1, further comprising:

   a plurality of transaction queues, associated with said plurality of MACs, said local bus interface, and said bus router, coupled to said transaction switch, for storing said transactions.

3. (previously presented) The integrated circuit of claim 1, wherein said memory comprises a random access memory.

4. (previously presented) The integrated circuit of claim 1, wherein said transaction switch comprises:

a buffer manager, for allocating portions of said memory to said plurality of MACs, said local bus interface, and said bus router, for buffering said data received thereby.

5. (previously presented) The integrated circuit of claim 4, wherein said buffer manager performs said allocating in an as-needed manner.

6-8. (canceled)

9. (previously presented) The integrated circuit of claim 1, wherein said local bus interface comprises a PCI interface.

10. (previously presented) The integrated circuit of claim 1, wherein said transaction switch is configured to receive a transaction posted by a first of said plurality of MACs in response to a packet received by said first of said plurality of MACs and to selectively switch said transaction to one of a second of said plurality of MACs and said bus router.

11. (original) The integrated circuit of claim 10, wherein said transaction switch selectively switches said transaction based on an InfiniBand destination local identification value included in said transaction.

12. (original) The integrated circuit of claim 11, wherein said transaction switch selectively switches said transaction to said bus router if an entry associated with said InfiniBand destination local identification value in a mapping table of said transaction switch indicates said transaction is destined for said bus router.

13. (previously presented) The integrated circuit of claim 12, wherein said transaction switch selectively switches said transaction to one of said plurality of MACs based on which of said plurality of MACs is associated with said InfiniBand destination local identification value in said mapping table if said entry indicates said transaction is not destined for said bus router.

14. (original) The integrated circuit of claim 11, wherein said first MAC parses said InfiniBand destination local identification value from said packet.

15. (original) The integrated circuit of claim 10, wherein said transaction includes an InfiniBand virtual lane number parsed from said packet.

16. (original) The integrated circuit of claim 10, wherein said transaction includes a destination queue pair number parsed from said packet.

17. (previously presented) The integrated circuit of claim 1, wherein said transaction switch is configured to receive a transaction posted by said bus router and to selectively switch said transaction to one of said plurality of MACs and said local bus interface.

18. (original) The integrated circuit of claim 17, wherein said transaction switch selectively switches said transaction based on a transaction type value included in said transaction.

19. (previously presented) The integrated circuit of claim 18, further comprising:

a second local bus interface, for performing addressed data transfers on a second local bus coupled thereto;

wherein said transaction switch selectively switches said transaction to one of said first and second local bus interfaces based on whether a local bus address included in said transaction falls into one or more predetermined address ranges of said first and second local buses.

20. (previously presented) The integrated circuit of claim 17, wherein said transaction includes an address in an address range of said local bus.

21. (previously presented) The integrated circuit of claim 1, further comprising:

a second local bus interface, coupled to said transaction switch, for performing addressed data transfers on a second local bus coupled thereto; and

a local bus bridge coupled between said first and second local bus interfaces for buffering data therebetween.

22. (previously presented) The integrated circuit of claim 1, further comprising:

a second local bus interface;

wherein said transaction switch is configured to receive a transaction posted by said first local bus interface in response to an addressed data transfer received by said first local bus interface and to switch said transaction to said second local bus interface.

23. (previously presented) A transaction switch for switching data between a plurality of data devices, comprising:

a memory, shared by the plurality of data devices for buffering data received thereby;

multiplexing logic, coupled to said memory, for controlling the transfer of data between the plurality of data devices and said memory; and

control logic, coupled to said multiplexing logic, for controlling said multiplexing logic;

wherein the plurality of data devices comprise a plurality of packetized data devices and a plurality of addressed data devices;

wherein said control logic is configured to selectively control said multiplexing logic to transfer data through said memory between two of said packetized data devices and between one of said packetized data devices and one of said addressed data devices;

wherein said control logic is configured to selectively control said multiplexing logic to transfer data through said memory between two of said packetized data devices and between one of said packetized data devices and one of said addressed data devices in response to a transaction posted to the transaction switch by the plurality of data devices;

wherein said transaction comprises a command to transfer data between said memory and one of the plurality of data devices;

wherein said transaction comprises an address of a buffer within said memory, wherein said data to be transferred in response to said command is stored in said memory.

24. (original) The transaction switch of claim 23, wherein said control logic is further configured to selectively control said multiplexing logic to transfer data through said memory between two of said addressed data devices.

25. (original) The transaction switch of claim 23, wherein said control logic is configured to selectively control said multiplexing logic to transfer data through said memory between two of said packetized data devices and between one of said packetized data devices and one of said addressed data devices concurrently.

26. (original) The transaction switch of claim 23, wherein at least two of said packetized data devices comprise InfiniBand interfaces.

27. (previously presented) The transaction switch of claim 23, wherein at least two of said addressed data devices comprise PCI interfaces.

28. (original) The transaction switch of claim 23, further comprising:

a buffer manager, for allocating portions of said memory to the plurality of data devices for buffering said data.

29. (original) The transaction switch of claim 28, wherein said buffer manager is configured to perform said allocating on substantially a first-come-first-serve basis.

30-32. (canceled)

33. (previously presented) The transaction switch of claim 23, wherein said transaction comprises an offset within said buffer for addressing portions of said data.

34. (previously presented) The transaction switch of claim 23, wherein said transaction comprises a tag for uniquely identifying said transaction from other transactions posted to the transaction switch by the plurality of data devices.

35. (original) The transaction switch of claim 23, wherein the plurality of data devices comprise a transport layer device, wherein the transaction switch is configured to receive transactions from said transport layer device for performing protocol translation of data between said one of said packetized data devices and said one of said addressed data devices.

36. (previously presented) A transaction switch for switching transactions and data between a plurality of data interfaces, the transaction switch comprising:

a memory, shared by the plurality of data interfaces, for buffering data received thereby;

a plurality of transaction queues, associated with each of the plurality of data interfaces, configured to store transactions, said transactions adapted to convey information to enable the plurality of data interfaces to transfer said data according to a plurality of disparate data transfer protocols supported thereby; and

control logic, coupled to said memory and said plurality of transaction queues, configured to route said data through said shared memory between the plurality of data interfaces and to switch said transactions between the plurality of data interfaces;

wherein at least one of the plurality of data interfaces comprises a bus router for performing a transport layer function between at least two of the plurality of data interfaces which support disparate data protocols;

wherein at least one of the plurality of data interfaces comprises a local bus interface for interfacing to a local bus, wherein at least one of the plurality of data interfaces comprises a MAC for interfacing to a network, wherein said bus router is configured to perform said transport layer function by writing packet header information into an allocated portion of said memory in front of data written into said allocated portion of said memory by said local bus interface;

wherein said bus router is configured to write said packet header information into said allocated portion of said memory before said MAC reads said packet from said allocated portion of said memory for writing to said network, without copying said data to another memory.

37. (original) The transaction switch of claim 36, wherein said control logic is configured to route said data between the plurality of data interfaces through said shared memory in response to said transactions received from said plurality of transaction queues.

38. (original) The transaction switch of claim 36, wherein at least a portion of said plurality of transaction queues is configured to store transactions adapted to convey information necessary to transfer data according to an InfiniBand protocol.

39. (previously presented) The transaction switch of claim 36, wherein at least a portion of said plurality of transaction queues is configured to store transactions adapted to convey information necessary to transfer data according to an PCI protocol.

40. (original) The transaction switch of claim 36, wherein said control logic is further configured to modify a transaction received from one of said plurality of transaction queues associated with a first of the plurality of data devices and to send said modified transaction to another one of said plurality of transaction queues.

41. (previously presented) An integrated circuit, comprising:

at least three data interfaces;

a memory, shared by said at least three data interfaces for buffering data therebetween; and

a transaction switch, coupled to said at least three data interfaces and said memory, for dynamically allocating portions of said memory to said at least three data interfaces for storing data therein, and for controlling access to said allocated portions of said memory by each of said at least three data interfaces;

wherein at least one of said at least three data interfaces is of a different type than the others;

wherein at least one of said at least three data interfaces comprises a bus router for performing a transport layer function between at least two other of said at least three data interfaces which support disparate data protocols;

wherein at least one of said at least three data interfaces comprises a local bus interface for interfacing to a local bus, wherein at least one of said at least three data interfaces comprises a MAC for interfacing to a network, wherein said bus router is configured to perform said transport layer function by writing packet header information into said allocated portions of said memory in front of data written into said allocated portions of said memory by said local bus interface;

wherein said bus router is configured to write said packet header information into said allocated portions of said memory before said MAC reads said packet from said allocated portion of said memory for writing to said network, without copying said data to another memory.

42. (original) The integrated circuit of claim 41, wherein at least one of said at least three data interfaces is a packetized data interface and at least one of said at least three data interfaces is an addressed data interface.

43. (original) The integrated circuit of claim 42, wherein said at least one packetized data interface is an InfiniBand interface.

44. (original) The integrated circuit of claim 42, wherein said at least one addressed data interface is a PCI interface.

45. (original) The integrated circuit of claim 41, wherein said transaction switch is configured to receive a transaction from a first of said at least three data interfaces and to selectively switch said transaction to one of another of said at least three data interfaces.

46. (original) The integrated circuit of claim 45, wherein said transaction is a packetized data transaction including packet destination information.

47. (original) The integrated circuit of claim 46, wherein said transaction switch is configured to selectively switch said packetized data transaction to said another of

said at least three data interfaces based on said packet destination information and information stored in a mapping table of said transaction switch.

48. (original) The integrated circuit of claim 47, wherein said transaction switch is configured to selectively switch said packetized data transaction to said another of said at least three data interfaces further based on information stored in a table mapping said packet destination information to said at least three data interfaces.

49. (original) The integrated circuit of claim 45, wherein in a first instance of said transaction said first and one of another of said at least three data interfaces are of a same type of interface, wherein in a second instance of said transaction said first and one of another of said at least three data interfaces are of a different type of interface.

50. (original) The integrated circuit of claim 49, wherein in said first instance each of said first and one of another of said at least three data interfaces is a packetized data interface type.

51. (original) The integrated circuit of claim 49, wherein in said second instance said first of said at least three data interfaces is a packetized data interface type and said one of another of said at least three data interfaces is an interface type capable of translating between packetized and addressed data.

52. (original) The integrated circuit of claim 49, wherein in said second instance said first of said at least three data interfaces is a packetized data interface type and said one of another of said at least three data interfaces is a transport level data interface.

53. (original) The integrated circuit of claim 45, wherein said transaction switch is configured to modify said transaction received from said first of said at least three data interfaces prior to selectively switching said received transaction to said one of another of said at least three data interfaces.

54. (original) The integrated circuit of claim 41, wherein said transaction switch is configured to receive a transaction from a first of said at least three data interfaces and to selectively switch said received transaction to two or more of another of said at least three data interfaces.

55. (original) The integrated circuit of claim 41, further comprising:

a plurality of transaction queues, coupled between said transaction switch and said at least three data interfaces, for storing transactions between said transaction switch and said at least three data interfaces.

56. (original) The integrated circuit of claim 55, further comprising a programmable register for specifying for at least a plurality of said plurality of transaction queues a number of transaction slots to be allocated for storing said transactions.

57-59. (canceled)

60. (original) The integrated circuit of claim 41, wherein a first of said at least three data interfaces is configured to post a transaction to said transaction switch for

instructing a second of said at least three data interfaces to transfer data to or from an offset in one of said allocated portions of said memory associated with a payload portion of a data packet.

61. (original) The integrated circuit of claim 41, wherein said transaction switch is further configured to de-allocate said portions of said memory.

62. (original) The integrated circuit of claim 41, wherein said transaction switch is configured to dynamically allocate said portions of said memory to said at least three data interfaces on a substantially as needed basis.

63-66. (canceled)

67. (currently amended) A network interface adapter, comprising:

a plurality of network ports, for interfacing to a network;

at least one addressed data bus interface;

a memory, for buffering data received by said plurality of network ports and said at least one addressed data bus interface;

a transport layer engine for performing a transport layer function for routing said data between said plurality of network ports and said at least one addressed data bus interface;

a plurality of transaction queues, associated with each of said plurality of network ports, said at least one addressed data bus interface and said transport layer engine, for storing transactions; and

a transaction switch, coupled to said plurality of transaction queues, configured to route said transactions between said plurality of network ports, said at least one addressed data bus interface and said transport layer engine;

~~a bus router for performing a transport layer function between said at least one addressed data bus interface and said plurality of network ports;~~

wherein said transport layer engine ~~bus router~~ is configured to perform said transport layer function by writing packet header information into an allocated portion of said memory in front of data written into said allocated portion of said memory by said at least one addressed data bus interface;

wherein said transport layer engine ~~bus router~~ is configured to write said packet header information into said allocated portion of said memory before one of said plurality of network ports reads said packet from said allocated portion of said memory for writing to said network, without copying said data to another memory.

68. (previously presented) The network interface adapter of claim 67, wherein at least a subset of said plurality of transaction queues comprise an input queue for said plurality of network ports to post said transaction to said switch.

69. (previously presented) The network interface adapter of claim 67, wherein said plurality of transaction queues comprise an output queue for said transaction switch to send said transaction to said plurality of network ports and said at least one addressed data bus interface.

70. (previously presented) The network interface adapter of claim 67, wherein said at least one addressed data bus interface is coupled to a data bus selected from a list comprising a Rapid I/O bus, a VESA bus, an ISA bus, a PCI bus, an LDT bus, an SDRAM bus, a DDR SDRAM bus, and a RAMBUS.

71-78. (canceled)

79. (currently amended) A transaction switch in a network device having a buffer memory and plurality of data devices, including packetized and addressed data devices, the transaction switch comprising:

a buffer manager, for allocating portions of the buffer memory to the plurality of data devices on an as-needed basis;

a plurality of data paths, coupling the buffer memory and the plurality of packetized and addressed data devices, for providing the plurality of data devices access to the buffer memory;

a mapping table, for storing packet destination identification information;

a plurality of transaction queues, for transferring transactions between the transaction switch and the plurality of data devices; and

control logic, coupled to said mapping table and said plurality of transaction queues, for selectively switching transactions between the plurality of data devices based on said mapping table information and based on contents of said transactions;

wherein at least one of the plurality of ~~data interfaces~~data devices comprises a bus router for performing a transport layer function between at least two of the plurality of ~~data interfaces~~data devices which support disparate data protocols;

wherein at least one of the plurality of ~~data interfaces~~data devices comprises a local bus interface for interfacing to a local bus, wherein at least one of the plurality of ~~data interfaces~~data devices comprises a MAC for interfacing to a network, wherein said bus router is configured to perform said transport layer function by writing packet header information into said allocated portions of the buffer memory in front of data written into said allocated portions of the buffer memory by said local bus interface;

wherein said bus router is configured to write said packet header information into said allocated portions of the buffer memory before said MAC reads said packet from said allocated portion of the buffer memory for writing to said network, without copying said data to another memory.

80. (previously presented) The transaction switch of claim 23, wherein a single integrated circuit comprises the transaction switch.

81. (previously presented) The transaction switch of claim 36, wherein a single integrated circuit comprises the transaction switch.

82. (previously presented) The network interface adapter of claim 67, wherein a single integrated circuit comprises the InfiniBand hybrid channel adapter/switch.

83. (previously presented) The integrated circuit of claim 1, wherein at least one of said plurality of MACs comprises an Infiniband MAC.

84. (previously presented) The integrated circuit of claim 1, wherein at least one of said plurality of MACs comprises an Ethernet MAC.

85. (previously presented) The transaction switch of claim 23, wherein at least one of said at least two of said packetized data devices comprises an Ethernet interface.

86. (previously presented) The transaction switch of claim 36, wherein at least a portion of said plurality of transaction queues is configured to store transactions adapted to convey information necessary to transfer data according to an Ethernet protocol.

87. (previously presented) The integrated circuit of claim 42, wherein said at least one packetized data interface is at least one Ethernet interface.

88-92. (canceled)

93. (previously presented) The integrated circuit of claim 1, wherein at least one of said plurality of local bus interfaces comprises an interface to a bus for coupling to a random access memory.

94. (previously presented) The transaction switch of claim 23, wherein at least one of said plurality of addressed data devices comprises an interface to a bus for coupling to a random access memory.

95. (previously presented) The transaction switch of claim 36, wherein at least one of said plurality of disparate data transfer protocols comprises a protocol for transferring data on a bus with a random access memory.

96. (previously presented) The integrated circuit of claim 41, wherein at least one of said at least three data interfaces comprises an interface to a bus for coupling to a random access memory.

97-99. (canceled)

100. (previously presented) A network interface adapter, comprising:

a local bus interface, configured to connect the adapter to a local bus and to perform an addressed data protocol on the local bus;

a media access controller (MAC), configured to connect the adapter to a network and to perform a packetized data protocol on the network;

a transport protocol engine, configured to perform protocol translation from the addressed data protocol to the packetized data protocol by creating a packet from data received on the local bus by the local bus interface;

a memory, directly accessible by each of the local bus interface, the MAC, and the transport protocol engine in a random access fashion;

a transaction switch, coupled to the local bus interface, the MAC, the transport protocol engine, and the memory, and configured to dynamically allocate a portion of the memory for buffering the data and the packet;

wherein the transport protocol engine is configured to perform the protocol translation before the MAC reads the packet from the memory for transmission on the network, wherein the transport protocol engine performs the protocol translation without copying the data to another memory;

wherein the transport protocol engine is further configured to perform second protocol translation from the packetized data protocol to the addressed data protocol by identifying a data payload within a second packet received from the network by the MAC and specifying a local bus address for the data payload;

wherein the transport protocol engine is further configured to perform the second protocol translation before the local bus interface reads the data payload from the memory for writing on the local bus to the local bus address, wherein the transport protocol engine performs the second protocol translation within the memory without copying the data to another memory.

101. (canceled)

102. (previously presented) The network interface adapter of claim 100, wherein a single integrated circuit comprises the local bus interface, the MAC, the transport protocol engine, the memory, and the transaction switch.

103. (previously presented) The network interface adapter of claim 100, wherein the transport protocol engine is configured to create the packet by writing a packet header into the memory in front of the data.

104. (previously presented) A network interface adapter, comprising:

a local bus interface, configured to connect the adapter to a local bus and to perform an addressed data protocol on the local bus;

a media access controller (MAC), configured to connect the adapter to a network and to perform a packetized data protocol on the network;

a transport protocol engine, configured to perform protocol translation from the packetized data protocol to the addressed data protocol by identifying a data payload within a packet received from the network by the MAC and specifying a local bus address for said data payload;

a memory, directly accessible by each of the local bus interface, the MAC, and the transport protocol engine in a random access fashion;

a transaction switch, coupled to the local bus interface, the MAC, the transport protocol engine, and the memory, and configured to dynamically allocate a portion of the memory for buffering the data payload and the packet;

wherein the transport protocol engine is configured to perform the protocol translation before the local bus interface reads the data payload from the memory for writing on the local bus to said local bus address, wherein the transport protocol engine performs the protocol translation without copying the data to another memory;

wherein the transport protocol engine is further configured to perform second protocol translation from the addressed data protocol to the packetized data protocol by creating a second packet from data received on the local bus by the local bus interface;

wherein the transport protocol engine is further configured to perform the second protocol translation before the MAC reads the second packet from the memory for transmission on the network, wherein the transport protocol engine performs the second protocol translation without copying the data to another memory.

105. (canceled)

106. (previously presented) The network interface adapter of claim 104, wherein a single integrated circuit comprises the local bus interface, the MAC, the transport protocol engine, the memory, and the transaction switch.

107. (previously presented) The network interface adapter of claim 104, wherein the transport protocol engine is configured to identify the data payload within the packet by specifying to the transaction switch an address of a location within the memory of the data payload.

108. (previously presented) A method for performing a transport layer function without double-buffering in a network interface adapter comprising a local bus interface, a media access controller (MAC), a transport protocol engine, and a memory managed by the transaction switch and directly accessible by each of the local bus interface, the MAC, and the transport protocol engine in a random access fashion, the method comprising:

allocating, by the transaction switch, a portion of the memory for buffering data received on the local bus by the local bus interface;

writing, by the local bus interface, the data into the allocated portion of the memory;

creating, by the transport protocol engine, with the data a packet within the allocated portion of the memory;

reading, by the MAC, the packet from the allocated portion of the memory for transmission on a network, after said creating, wherein the transport protocol engine performs said creating the packet without copying the data to another memory;

allocating, by the transaction switch, a second portion of the memory for buffering a second packet received from the network by the MAC;

writing, by the MAC, the second packet into the second allocated portion of the memory;

identifying, by the transport protocol engine, a data payload within the second packet;

specifying, by the transport protocol engine, a local bus address for the data payload; and

reading, by the local bus interface, the data payload from the second allocated portion of the memory for writing on the local bus to the local bus address, after said identifying and said specifying, wherein the transport protocol engine performs said identifying and said specifying without copying the data to another memory.

109. (canceled)

110. (previously presented) The method of claim 108, wherein said creating the packet comprises:

writing, by the transport protocol engine, a packet header into the allocated portion of the memory in front of the data.

111. (previously presented) A method for performing a transport layer function without double-buffering in a network interface adapter comprising a local bus interface, a media access controller (MAC), a transport protocol engine, and a memory managed by a transaction switch and directly accessible by each of the local bus interface, the MAC, and the transport protocol engine in a random access fashion, the method comprising:

allocating, by the transaction switch, a portion of the memory for buffering a packet received from the network by the MAC;

writing, by the MAC, the packet into the allocated portion of the memory;

identifying, by the transport protocol engine, a data payload within the packet;

specifying, by the transport protocol engine, a local bus address for the data payload;

reading, by the local bus interface, the data payload from the allocated portion of the memory for writing on the local bus to the local bus address, after said identifying and said specifying, wherein the transport protocol engine performs said identifying and said specifying without copying the data to another memory;

allocating, by the transaction switch, a second portion of the memory for buffering data received on the local bus by the local bus interface;

writing, by the local bus interface, the data into the second allocated portion of the memory;

creating, by the transport protocol engine, with the data a second packet within the second allocated portion of the memory; and

reading, by the MAC, the second packet from the second allocated portion of the memory for transmission on a network, after said creating, wherein the transport protocol engine performs said creating the second packet without copying the data to another memory.

112. (canceled)

113. (previously presented) The method of claim 111, wherein said identifying comprises:

specifying, by the transport protocol engine, to the transaction switch an address of a location within the memory of the data payload.